DOI:
10.1007/978-3-319-40379-3_32

Document Version
Peer reviewed version

Citation for published version (APA):
Schneider, E., Sklar, E., & Parsons, S. (2016). Evaluating Multi-Robot Teamwork in Parameterised Environments. In Towards Autonomous Robotic Systems: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). (Vol. 9716, pp. 301-313). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 9716). SpringerVerlag Berlin Heidelberg. 10.1007/978-3-319-40379-3_32

Evaluating Multi-Robot Teamwork in Parameterised Environments

Eric Schneider[1], Elizabeth I. Sklar[2], and Simon Parsons[2]

[1] Dept of Computer Science, University of Liverpool, UK
eric.schneider@liverpool.ac.uk
[2] Dept of Informatics, King's College London, UK
{elizabeth.sklar,simon.parsons}@kcl.ac.uk

Abstract. The work presented here investigates the impact of certain environmental parameters on the performance of a multi-robot team conducting exploration tasks. Experiments were conducted with physical robots and simulated robots, and a diverse set of metrics was computed. The experiments were structured to highlight several factors: (a) single-robot versus multi-robot tasks; (b) independent versus dependent (or "constrained") tasks; and (c) static versus dynamic task allocation modes. Four different task allocation mechanisms were compared, in two different exploration scenarios, with two different starting configurations for the robot team. The results highlight the distinctions between parameterised environments (characterised by the factors above, the robots' starting positions and the exploration scenario) and the effectiveness of each task allocation mechanism, illustrating that some mechanisms perform statistically better in particular environment parameterisations.

Keywords: Multi-robot team, auction mechanism, task allocation

1 Introduction

A future is envisioned in which autonomous mobile robots work together across a wide range of scenarios, from disaster response to humanitarian de-mining to factory maintenance. As the set of possible environments expands, so do the demands for multi-robot teamwork, requiring robots to operate in increasingly complex and challenging settings. A key requirement for deployment of such teams is a comprehensive understanding of the factors that contribute to these challenges. For example, is it important that the team knows what tasks it will be asked to complete beforehand? Do dependencies between tasks contribute to the efficacy of a task allocation? If some tasks require two robots to work together, how does that impact the organisation of the team? These questions form an open area of research, which falls under the heading of *multi-robot task allocation*: given a mission composed of a number of tasks, what is the best way to assign tasks to robots so that the mission is executed in an efficient way according to some performance metric(s) (e.g., minimising distance travelled)?

There are various approaches to multi-robot task allocation, ranging from centralised and fully connected (where a single controller talks to and coordinates all robots) to distributed and partially connected (where communication is limited [2]). *Market-based approaches*, which use estimates of cost or utility to distribute tasks, as goods are priced and distributed in an economic market, fall somewhere in the middle of this range. *Auctions* are a popular form of market for task allocation. Tasks are advertised to team members, who then bid on them using private valuation functions. Many different auction mechanisms (i.e., the rules and procedures that describe how, when and to whom tasks are advertised, bid upon and awarded) have been studied [3].

Most of the existing work studies environments in which tasks are known ahead of time, are independent, can be completed in any order, and require only one robot. Existing taxonomies [6, 16] suggest three task dimensions, labelling well-studied environments as *single-robot* (*SR*), *independent* (IT) and *static* (*SA*). In our work, we are investigating task allocation in more complex environments using several auction-based mechanisms found in the literature. Our earlier work evaluated *static* versus *dynamic* task allocation factors, comparing situations where tasks were all known ahead of time and were allocated before execution of any task commenced (*SR-IT-SA* [21, 24]) and situations where tasks appeared *during* execution, meaning that allocation occurred dynamically, after some tasks had commenced (*SR-IT-DA* [25]). Here, we consider two additional confounding factors: *multi-robot* (*MR*) tasks, where more than one robot is required (e.g., moving a heavy object); and *constrained* (CT) tasks, where a task may be dependent on others to be completed before it can be executed (e.g., clearing debris from a doorway before being able to enter a room).

Our long term goal is to develop a comprehensive understanding of the factors that contribute to multi-robot team performance in varied environments. Here, we test two hypotheses. The first hypothesis is that *within* a single environment, the different mechanisms evaluated here produce statistically significantly different results, according to particular performance metrics. Thus, *for any one point* in the environment landscape, we can identify one task allocation mechanism that reliably performs the best for a given metric. The second hypothesis is that *across* multiple environments, there is no definitive or consistent ranking of mechanisms across the metrics. Thus, *across all points* in the environment landscape, none of the task allocation mechanisms evaluated performs the best for a given metric. As shown here, we have proven both these hypotheses through empirical results obtained on physical robots, backed up with results obtained in simulation experiments.

2 Related work

The use of market mechanisms in distributed computing can be considered to have begun with Smith's *contract net* protocol [27], and this was followed by Wellman and Wurman's *market-aware agents* [29]. A primary strength of market-based approaches is their reliance only on local information and/or the self-interest of agents to arrive at efficient solutions to large-scale, complex problems that are otherwise intractable [3]. The most common instantiations of market mechanisms in multi-robot systems are *auctions.* Auctions are commonly used for distribution tasks, where resources or roles are treated as commodities and auctioned to agents. Existing work analyzes the effects of different auction mechanisms [1, 3, 13, 30], bidding strategies [23], dynamic task re-allocation [7] and levels of commitment to the contracts [18] on the overall solution quality.

In domains where there is a strong synergy between items, single-item auctions can result in sub-optimal allocations [1]. In multi-robot exploration, studied here, strong synergies may exist between tasks. Combinatorial auctions remedy this limitation by allowing agents to bid on bundles of items, and minimise the total travel distance because they take the synergies between tasks into account [10]. Combinatorial auctions suffer, however, from the computational costs of bid generation and bundle valuation by the agents, and winner determination by the auction mechanism itself, all of which are NP-hard [11]. As a result, a body of work has grown up around the *sequential single-item auction* (SSI) [11], which has been proven to create close to optimal allocations, handling synergies while not suffering from the complexity issues of combinatorial auctions.

This paper is a further contribution to the body of work around SSI, extending the use of SSI and related mechanisms to task environments that are, according to the taxonomies developed by Gerkey and Mataríc [6] and Landén et al. [16]: *multi-robot* (*MR*), *constrained* (*CT*) and *dynamic* (*DA*). Auction-based approaches to task allocation have been proposed for tasks with precedence [17], with temporal [8, 20] constraints, and for dynamic environments [25] with single robot tasks. Environments that contain multi-robot tasks, with and without constraints, are less well investigated than their single-robot counterparts [12].

Of all the literature on auctions in multi-robot teams, [19] and [26] are the most closely related to our work. Both evaluate SSI in simulation and so one could argue that their work evaluates the practical cost of solutions generated using SSI. However, the focus of both [19] and [26] is on finding optimal mechanisms for dynamic task reallocation during execution rather than being concerned with the quality of team performance across a full set of tasks, which is our focus. In addition, neither [19] nor [26] consider the range of metrics that we do and so are unable, for example, to comment on the load balancing that our measurement of "idle time" exposes or the amount of time one robot waits for another to arrive at a joint location so they can execute a multi-robot task together.

3 Methodology

Formal description. We extend the notation of [11], where a set of n **robots**, $R = \{r_0, \ldots, r_{n-1}\}$, forms a *team*; a set of m **tasks**, $T = \{t_0, \ldots, t_{m-1}\}$, comprises a *mission*; and $T(\rho)$ is the set of tasks assigned to robots $\rho \in R$. We make three extensions.

First, we specify dependencies between tasks in T. Following [16], *Independent Tasks (IT)* can be executed in any order, whereas *Constrained Tasks (CT)* have an implicit ordering. For example, suppose that t_p is a task to clear debris blocking the entrance to a passageway and t_q is a task to collect sensor data inside that passageway: t_p must be completed *before* t_q can proceed. Formally: the set of *constrained tasks* CT is a set of pairs of tasks (t_p, t_q), $t_p, t_q \in T$, such that t_p must be completed before t_q can proceed. The set of relations $(t_p, t_q) \in CT$ defines a partial order over T.

Second, we specify the number of robots required to execute each task. Following [6], *Single-Robot (SR)* tasks need only be assigned to one robot, whereas *Multi-Robot (MR)* tasks need more than one robot. Formally, each task, t, has an associated value $t.req$ defining the number of robots required to complete that task; thus if $t.req = 1$, then t is an SR task. If $t.req > 1$, then t is an MR task and $t \in T(\rho)$ where $\rho \subseteq R$ such that $|\rho| = t.req$.

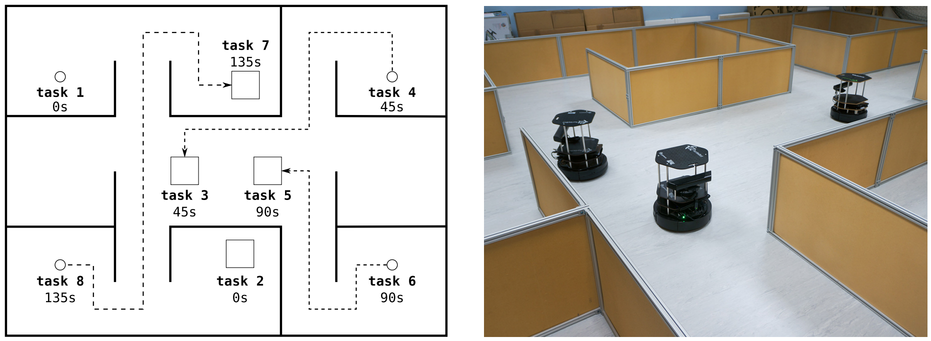
Third, we specify the *arrival time* of each task. Each task t has an associated value $t.arr$ which defines the time, τ, at which any of the robots $\rho \subseteq R$ become aware of t. Following [16], we distinguish between *Static Allocation (SA)* environments, where every $t.arr$ time is before the execution of any task begins, and *Dynamic Allocation (DA)* environments, where $t.arr$ values may occur after the execution of at least one task begins. Each mission consists of an initial step where tasks are announced; next, tasks are assigned to robots; and then, tasks are executed. In an SA environment, the allocation of all tasks in the mission occurs before any task is executed, whereas, in DA, the steps may overlap.

Previous work has experimentally evaluated auction-based mechanisms for task allocation in SR-IT-SA [11] and SR-IT-DA [25] settings. Korsah et al. [12] provide a comprehensive discussion of prior work in this domain, across a range of task allocation methodologies (not just auction-based). In the work presented here, we focus on dynamic allocation environments and present experimental evaluations of MR-CT-DA, MR-IT-DA, SR-CT-DA and SR-IT-DA.

Metrics. To evaluate the performance of a team, we consider metrics that measure the performance of both individual robots and the team as a whole. In any work with robots, power consumption is the fundamental scarce resource that a robot possesses. Robot batteries only last for a limited time, and so, all other things being equal, we prefer task allocations and subsequent executions that minimise battery usage. As in [10, 11, 14, 15, 28], therefore, we measure the **distance travelled** by the robots in executing a set of tasks—both individually and as a group—since this is a suitable proxy for power consumption.[3]

Time is also important, which we measure in several ways: **run time** is the time between the start of an experiment and the point at which the last robot on the team completes the tasks allocated to it; **deliberation time** is a component of run time, the time that it takes for tasks to be allocated amongst the robots; **execution time** is another component of run time, the time it takes robots to complete tasks once they have been allocated; **movement time** is the

[3] Note that we compute distance not by looking at the shortest distances between the task locations, but is (as closely as we can establish) the actual distance travelled by the robots during task execution. We collect frequent position updates, compute the Euclidean distance between successive positions, and sum these.



(a) MR-CT-DA Scenario

(b) The physical arena

Fig. 1. A MR-CT task scenario and lab in which physical experiments were conducted. Bold lines indicate the walls of the arena. Task locations are shown as circles for single-robot tasks or squares for two-robot tasks. A dashed line from task t_p to task t_q indicates a precedence constraint (t_p, t_q). Arrival times for each task $(t.arr)$ are also shown.

time robots spend actually moving, without interruption, toward task locations; **delay time** is the amount of time robots spend avoiding collisions with each other; **waiting time** is the amount of time robots wait for others to arrive at MR task locations; and **idle time** is the amount of time that robots wait for others to complete tasks, having completed their own.

Mechanisms. Our experiments employ four task allocation mechanisms:
(1) In round-robin (RR), tasks (T) and robots (R) are placed in two ordered lists. The first task, t_0, is allocated to the first robot, r_0. If $t_i.req > 1$ (an MR task), then it is also allocated to the next robot and so on until $t_i.req$ is reached. If $t_i.req = 1$ (an SR task), then the next task, t_{i+1}, is allocated to the next robot.
(2) In ordered single-item auction (OSI) [21], the tasks are placed in an ordered list. Each task t_i in turn is advertised to all the robots. Each robot makes a bid for the task, where the bid value is the increase in the total path cost that the robot estimates (using A*[9]) it would incur if it were to win that task. The task is allocated to the $t_i.req$-lowest bidding robots and the next task is auctioned.
(3) In sequential single-item auction (SSI) [11], unallocated tasks are presented to all robots simultaneously. Each robot bids on the task with the lowest cost (computed as in OSI), and the task with the lowest bid is allocated to the robot that made the bid. If the winning task, t_w, has been completely allocated (to $t_w.req$ robots), it is removed from the set of tasks to be advertised in the next round and the process is repeated until all tasks have been allocated.
(4) In parallel single-item auction (PSI) [11], allocation starts like SSI: all robots bid on all tasks from their current locations. All the tasks are allocated in one round, with each task t_i going to the $t_i.req$-lowest bidding robots that bid on it.

4 Experiments

We conducted a series of experiments comparing the task allocation mechanisms described earlier in a structured set of $\langle SR|MR\rangle\langle IT|CT\rangle\langle SA|DA\rangle$ environments. Here we describe the system used to conduct these experiments, the specific scenarios evaluated, and our results.

System Description. Task allocation is conducted by a central *auctioneer* agent, which communicates the start of an auction and awards tasks to *robot controller* agents.[4] Each robot controller submits bids, the auctioneer determines the winner(s) of the auction and allocates tasks accordingly. Robot controllers then execute tasks autonomously. Our software architecture is agnostic about whether the team executes its tasks on real robot hardware or in simulation. Our physical platform is the Turtlebot2,[5] which has a differential drive base and a colour/depth-sensing Microsoft Kinect camera. The ROS [22] navigation stack provides communication, localisation and path planning capabilities. Our simulated robot (using Stage [5]) has the same properties as its physical counterpart.

The operating environment for our robots is an office-like setting with rooms opening off a central hallway. The layout of this environment is shown in Figure 1(a), and a photograph is given in Figure 1(b). While this is a smaller environment than that studied by some others (e.g., [11]), our setup allows us to run parallel experiments on physical robots and—on a larger scale—in simulation, which produces more statistically significant results.

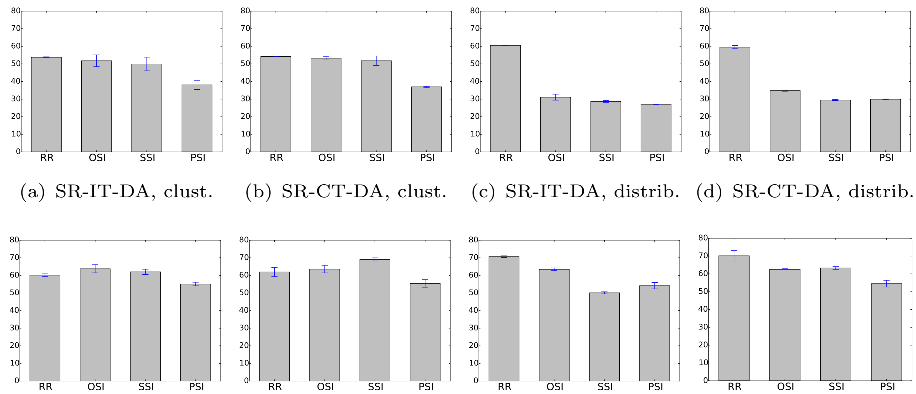
Experimental Setup. An experimental condition is defined by the starting locations of the robots and the task scenario (defined by task locations, task arrival times, constraints and robot requirements). This work investigates routing tasks—a robot executes a task simply by driving to the task's location. All of the experiments reported here involve a team of $n = 3$ robots. We used two sets of *starting configurations* for the robot team: one *clustered* the robots in the "room" in the lower left corner of the arena, while the other *distributed* the robots at three corners of the map. We examined four different *parameterised environments*, all with dynamic allocation (DA), combining single-robot (SR) vs. multi-robot (MR) and independent (IT) vs. constrained (CT) tasks: *SR-IT-DA, SR-CT-DA, MR-IT-DA* and *MR-CT-DA*. We employed two different *task scenarios*. Figure 1(a) shows a diagram of the first task scenario.

The aim in choosing this combination of parameterised environments was to see how performance of the four task allocation mechanisms varied along the MR/SR and CT/IT dimensions. In total, 192 physical and 960 simulation trials were performed: 2 *starting configurations* $\times$ 4 *parameterised environments* $\times$ 2 *task scenarios* $\times$ 4 *allocation mechanisms* $\times$ $\{3$ *physical* $|$ 15 *simulation*$\}$ *trials*. For each experiment, we recorded the metrics described in Section 3.

Results. Figures 2–3 and Table 1 contain representative results from the experiments. Figure 2 shows the average *distance travelled* by the team in eight

[4] Though bidding and winner determination are managed centrally, there is no centralised control in the usual sense. The auction could also be distributed among the robots as in [4].

[5] www.turtlebot.com



(a) SR-IT-DA, clust. (b) SR-CT-DA, clust. (c) SR-IT-DA, distrib. (d) SR-CT-DA, distrib.

(e) MR-IT-DA, clust. (f) MR-CT-DA, clust. (g) MR-IT-DA, distrib. (h) MR-CT-DA, distrib.

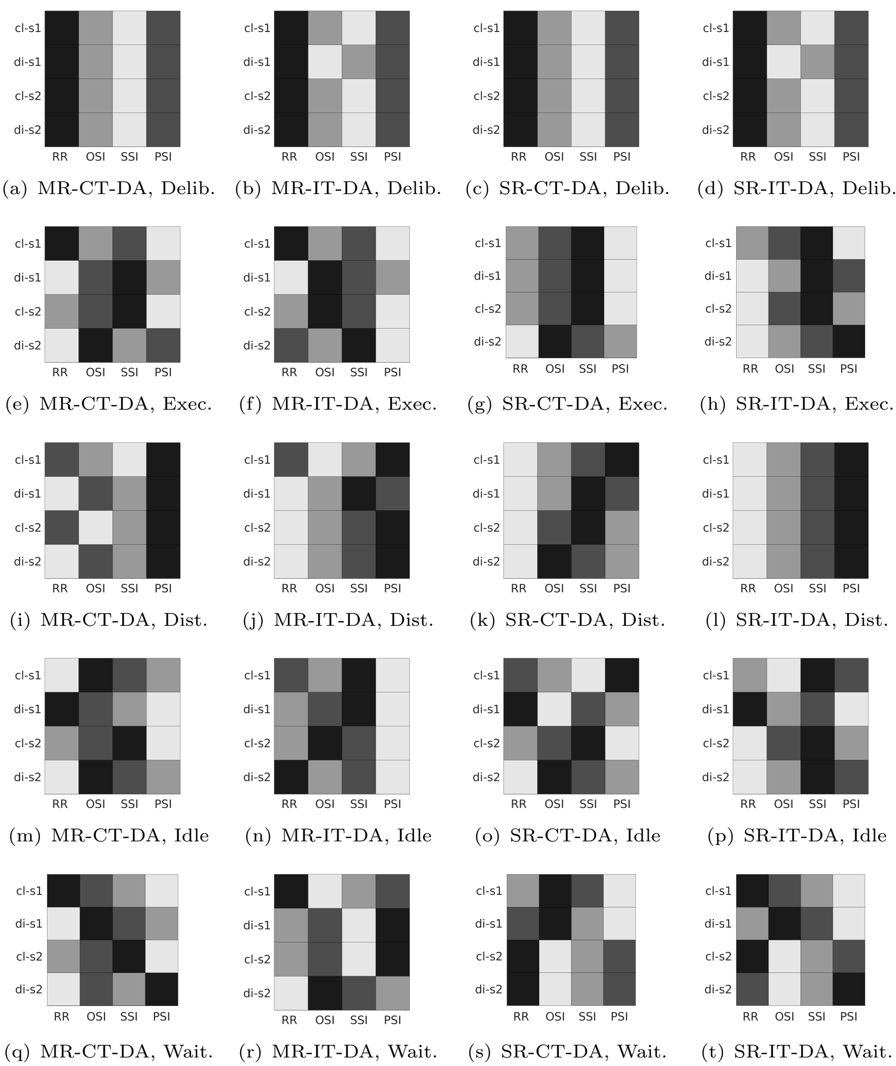
Fig. 2. Average distance (meters) travelled in physical experiments for variations of the scenario shown in Figure 1(a). Mechanisms are ordered RR, OSI, SSI and PSI.

variations of the scenario shown in Figure 1(a). In each plot, average travel distances resulting from allocations produced by RR, OSI, SSI, and PSI are shown from left to right. Error bars indicate 95% confidence intervals. Figures 2(a) and 2(b) show how, in the SR-clustered conditions of this scenario, PSI allocations result in distances that are significantly shorter than those produced by the other mechanisms. As we move to distributed-start conditions of the scenario (Figures 2(c) and 2(d)), differences among three of the mechanisms diminish but remain statistically significantly different, while RR continues to lead to dramatically longer distances. This result is similar to those reported in [25], where it was shown that a starting configuration that distributes team members more evenly amongst tasks tends to lessen the advantages of mechanisms such as SSI that exploit clustering properties of task locations. In MR conditions of the same scenario, the results are somewhat different. For example, RR doesn't always result in the longest distances nor does PSI always result in the shortest. The relative rankings of the mechanisms are much less predictable than in the SR conditions. Our second experimental task scenario produced similar results.

We can choose other of our performance metrics to examine individually. But with nine metrics and a large number of combinations of environments and experimental configurations, we want to make sense of the results as a whole. Do any of the mechanisms produce the best performance across environments or experimental configurations? Do clear patterns emerge? We address these questions in the following section by examining the data in aggregate.

5 Analysis

Here, we focus on five different performance metrics. *Deliberation time* is a component of overall run time and a good measure of how well an allocation mecha-



(a) MR-CT-DA, Delib. (b) MR-IT-DA, Delib. (c) SR-CT-DA, Delib. (d) SR-IT-DA, Delib.

(e) MR-CT-DA, Exec. (f) MR-IT-DA, Exec. (g) SR-CT-DA, Exec. (h) SR-IT-DA, Exec.

(i) MR-CT-DA, Dist. (j) MR-IT-DA, Dist. (k) SR-CT-DA, Dist. (l) SR-IT-DA, Dist.

(m) MR-CT-DA, Idle (n) MR-IT-DA, Idle (o) SR-CT-DA, Idle (p) SR-IT-DA, Idle

(q) MR-CT-DA, Wait. (r) MR-IT-DA, Wait. (s) SR-CT-DA, Wait. (t) SR-IT-DA, Wait.

Fig. 3. Heat maps for the physical experiment data on each parameterised environment. Each heatmap shows the two different scenarios and two different experimental conditions. For a given scenario/condition pair (row) the colour of the squares indicates the rank order of the mechanism (column). The darkest square indicates the lowest value of the metric (best mechanism), the lightest square indicates the highest value (worst mechanism). (a)–(d) show deliberation time, (e)–(h) show execution time, (i)–(l) show distance, (m)–(p) show idle time, and (q)–(t) show waiting time.

(a) Deliberation time

Physical						Simulation					
	$F(3,8)$	p		$F(3,8)$	p		$F(3,56)$	p		$F(3,56)$	p
MR-CT-DA-			SR-CT-DA-			MR-CT-DA-			SR-CT-DA-		
cl-s1	83.96	0.010	cl-s1	71.77	0.010	cl-s1	28709.89	0.010	cl-s1	241.51	0.010
di-s1	158.13	0.010	di-s1	43.87	0.010	di-s1	54561.93	0.010	di-s1	213.79	0.010
cl-s2	3901.58	0.010	cl-s2	1766.23	0.010	cl-s2	18630.69	0.010	cl-s2	30977.14	0.010
di-s2	3080.90	0.010	di-s2	3708.91	0.010	di-s2	22404.35	0.010	di-s2	21734.58	0.010
MR-IT-DA-			SR-IT-DA-			MR-IT-DA-			SR-IT-DA-		
cl-s1	93.79	0.010	cl-s1	5038.94	0.010	cl-s1	24591.32	0.010	cl-s1	174.05	0.010
di-s1	1150.34	0.010	di-s1	45.53	0.010	di-s1	44307.68	0.010	di-s1	2089.02	0.010
cl-s2	5124.26	0.010	cl-s2	37639.65	0.010	cl-s2	15842.79	0.010	cl-s2	23317.28	0.010
di-s2	5364.80	0.010	di-s2	146.10	0.010	di-s2	44591.27	0.010	di-s2	27112.49	0.010

(b) Execution time

	$F(3,8)$	p		$F(3,8)$	p		$F(3,56)$	p		$F(3,56)$	p
MR-CT-DA-			SR-CT-DA-			MR-CT-DA-			SR-CT-DA-		
cl-s1	1.39	**0.950**	cl-s1	19.70	0.010	cl-s1	30.43	0.010	cl-s1	60.39	0.010
di-s1	9.58	0.010	di-s1	3.27	**0.950**	di-s1	5.94	0.010	di-s1	22.79	0.010
cl-s2	5.49	0.050	cl-s2	19.72	0.010	cl-s2	24.02	0.010	cl-s2	19.82	0.010
di-s2	3.19	**0.950**	di-s2	24.63	0.010	di-s2	9.88	0.010	di-s2	39.51	0.010
MR-IT-DA-			SR-IT-DA-			MR-IT-DA-			SR-IT-DA-		
cl-s1	2.82	**0.950**	cl-s1	18.58	0.010	cl-s1	36.39	0.010	cl-s1	33.09	0.010
di-s1	1.54	**0.950**	di-s1	11.17	0.010	di-s1	9.53	0.010	di-s1	14.14	0.010
cl-s2	3.92	**0.950**	cl-s2	9.93	0.010	cl-s2	6.62	0.010	cl-s2	28.28	0.010
di-s2	0.77	**0.950**	di-s2	79.93	0.010	di-s2	5.10	0.010	di-s2	15.93	0.010

(c) Distance travelled

	$F(3,8)$	p		$F(3,8)$	p		$F(3,56)$	p		$F(3,56)$	p
MR-CT-DA-			SR-CT-DA-			MR-CT-DA-			SR-CT-DA-		
cl-s1	7.76	0.010	cl-s1	30.83	0.010	cl-s1	35.88	0.010	cl-s1	312.84	0.010
di-s1	13.04	0.010	di-s1	784.63	0.010	di-s1	4817.66	0.010	di-s1	75593.00	0.010
cl-s2	12.90	0.010	cl-s2	7.70	0.010	cl-s2	33.75	0.010	cl-s2	60.12	0.010
di-s2	9.39	0.010	di-s2	996.79	0.010	di-s2	132.54	0.010	di-s2	1395.83	0.010
MR-IT-DA-			SR-IT-DA-			MR-IT-DA-			SR-IT-DA-		
cl-s1	10.38	0.010	cl-s1	6.01	0.050	cl-s1	390.48	0.010	cl-s1	436.66	0.010
di-s1	68.46	0.010	di-s1	173.25	0.010	di-s1	3121.61	0.010	di-s1	98521.39	0.010
cl-s2	13.30	0.010	cl-s2	29.16	0.010	cl-s2	122.99	0.010	cl-s2	231.39	0.010
di-s2	10.21	0.010	di-s2	2823.98	0.010	di-s2	527.39	0.010	di-s2	3676.25	0.010

(d) Total idle time

	$F(3,8)$	p		$F(3,8)$	p		$F(3,56)$	p		$F(3,56)$	p
MR-CT-DA-			SR-CT-DA-			MR-CT-DA-			SR-CT-DA-		
cl-s1	0.72	**0.950**	cl-s1	8.44	0.010	cl-s1	40.63	0.010	cl-s1	40.08	0.010
di-s1	2.17	**0.950**	di-s1	4.33	0.050	di-s1	36.85	0.010	di-s1	34.25	0.010
cl-s2	8.28	0.010	cl-s2	6.23	0.050	cl-s2	112.31	0.010	cl-s2	7.00	0.010
di-s2	4.30	0.050	di-s2	29.89	0.010	di-s2	70.23	0.010	di-s2	29.02	0.010
MR-IT-DA-			SR-IT-DA-			MR-IT-DA-			SR-IT-DA-		
cl-s1	111.22	0.010	cl-s1	2.19	**0.950**	cl-s1	117.09	0.010	cl-s1	14.87	0.010
di-s1	7.90	0.010	di-s1	6.69	0.050	di-s1	40.33	0.010	di-s1	52.47	0.010
cl-s2	20.62	0.010	cl-s2	4.12	0.050	cl-s2	99.37	0.010	cl-s2	12.82	0.010
di-s2	16.53	0.010	di-s2	90.31	0.010	di-s2	16.58	0.010	di-s2	8.40	0.010

(e) Total waiting time

	$F(3,8)$	p		$F(3,8)$	p		$F(3,56)$	p		$F(3,56)$	p
MR-CT-DA-			SR-CT-DA-			MR-CT-DA-			SR-CT-DA-		
cl-s1	26.38	0.010	cl-s1	100.07	0.010	cl-s1	10.02	0.010	cl-s1	1260.61	0.010
di-s1	1.28	**0.950**	di-s1	9.19	0.010	di-s1	30.23	0.010	di-s1	100.39	0.010
cl-s2	0.15	**0.950**	cl-s2	6.01	0.050	cl-s2	16.90	0.010	cl-s2	38.08	0.010
di-s2	8.92	0.010	di-s2	22.55	0.010	di-s2	20.93	0.010	di-s2	6.94	0.010
MR-IT-DA-			SR-IT-DA-			MR-IT-DA-			SR-IT-DA-		
cl-s1	4.21	0.050	cl-s1	0.25	**0.950**	cl-s1	28.44	0.010	cl-s1	0.63	**0.950**
di-s1	0.26	**0.950**	di-s1	0.42	**0.950**	di-s1	23.39	0.010	di-s1	0.64	**0.950**
cl-s2	0.30	**0.950**	cl-s2	2.49	**0.950**	cl-s2	14.05	0.010	cl-s2	0.00	**0.950**
di-s2	1.00	**0.950**	di-s2	1.69	**0.950**	di-s2	8.03	0.010	di-s2	1.32	**0.950**

Table 1. F-ratios for 5 different metrics

nism scales with the number of tasks and the size of the team. *Execution time* is another component of run time and one of the main measures we would like to minimise, the other being *distance*. We also look at *idle time* as a measure of how well balanced the task load is among the team. Finally, we look at *waiting time*. This is a feature specific to the MR and CT environments. A key contribution of our work is extending experimental results, particularly with physical robots, into MR and CT environments.

One of our long term goals is to develop task allocation mechanisms, or methods of choosing mechanisms, that perform well in different environments. Underlying this is the assumption that some mechanisms lead to better performance outcomes in some environments than others, and that there may not be a single mechanism that is best suited for all environments. We suggest two research hypotheses to evaluate this assumption and use the results of experiments discussed here to provide evidence for them.

The first hypothesis is that *within* a single $\langle sc, pe, ts \rangle$ tuple (where sc =starting configuration, pe =parameterised environment, and ts =task scenario), the four mechanisms examined here produce statistically significantly different results, according to our performance metrics. It is important to show that performance differences between mechanisms exist in the first place before examining the effects of varying environments. To evaluate this first hypothesis, we apply *analysis of variance (ANOVA)* to determine if there are significant differences between the different mechanisms. We ran ANOVA on the four samples—one for each mechanism in each $\langle sc, pe, ts \rangle$ tuple. If the null hypothesis were true and the differences among the four samples were due to chance, then the likelihood of producing the F-ratio would be less than $p\%$. The F-ratios of samples from both physical and simulation experiments are shown in Table 1. These F-ratios (p-value $= 0.01$) indicate a significant performance difference between the populations (mechanisms). For example, in the case of deliberation time (Table 1(a)), very large F-ratio values are the result of comparing RR, a simple mechanism that runs very quickly, with the others. In contrast, F-ratios for distance travelled (Table 1(c)) are lower but still above the critical value for the significance level and degrees of freedom tested. This supports our first hypothesis.

The second hypothesis is that *across* multiple $\langle sc, pe, ts \rangle$ tuples, there is no definitive ranking amongst the metrics for each mechanism. Figure 3 shows performance rankings obtained from physical experiments in the form of heat maps. Each row of heat maps in the figure corresponds to one of the five metrics discussed above. Within each heat map, the four columns correspond to the four task allocation mechanisms (RR, OSI, SSI, PSI, from left to right). The rows of each heat map are labelled with a variation of a particular scenario. For example, *cl-s1* indicates *clustered, scenario 1*. The shading of a cell indicates its rank: darker shades indicate lower values for that metric. While the ANOVA results mentioned in support of the first hypothesis don't directly measure the degree to which any pair of mechanisms differed in performance, they do provide evidence that the rankings shown in the heat maps are based on statistically significant differences. The heat maps for *deliberation time* (Figure 3(a)–3(d)) reveal some consistency when comparing environments and experimental conditions (rows within a single heat map, and across heat maps in the same row of the figure). RR is always the quickest to run, followed by PSI, while OSI and SSI trade ranks depending on the experimental condition. Apart from deliberation time, this type of performance ranking does not hold in a consistent way for the other metrics when comparing across environments and experimental conditions. This supports our second hypothesis. Our next steps involve looking at more environments and experimental conditions that vary in systematic ways, to help discover correspondences between parameters of environments and the performance characteristics of different task allocation mechanisms. The type of analysis presented here is likely to be a useful tool for this future endeavour.

6 Summary

The work presented here tests two hypotheses: (1) within a single parameterised environment, a given task allocation mechanism can be proven to consistently outperform others for certain metrics; and (2) across a varied set of parameterised environments, no single task allocation mechanism will consistently outperform others for any metrics. We conducted experiments with physical robots, as well as simulated robots in an environment that parallels our physical setup. Our empirical results support both of these hypotheses.

Future work will involve exploration of more complex scenarios in simulation, with larger teams and more tasks, as well as additional experiments with physical robots. In addition, we will be assessing how *task duration* affects the metrics presented here, by varying the time it takes to complete different tasks. Our long term goal is to identify the features of parameterised environments and/or task scenarios that influence the performance of the different mechanisms, so that the differences in rankings highlighted here can be attributed to particular features of a given experimental environment.

References

1. M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. M. Griffin, and A. Kleywegt. Robot Exploration with Combinatorial Auctions. In *IROS*, 2003.
2. E. Bonabeau, M. Dorigo, and G. Théraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.
3. M. B. Dias, R. Zlot, N. Kalra, and A. Stentz. Market-based multirobot coordination: A survey and analysis. *Proc of the IEEE*, 94(7), 2006.
4. P. Ezhilchelvan and G. Morgan. A dependable distributed auction system: Architecture and an implementation framework. In *Int. Symp. Aut. Decent. Sys.*, 2001.
5. B. Gerkey, R. T. Vaughan, and A. Howard. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In *Int. Conf. Adv. Robotics*, 2003.
6. B. P. Gerkey and M. J. Mataríc. A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. J. Robotics Research*, 23(9), 2004.
7. M. Golfarelli, D. Maio, and S. Rizzi. A task-swap negotiation protocol based on the contract net paradigm. TR-005-97, DEIS/CSITE/Università di Bologna, 1997.

8. M. Gombolay, R. Wilcox, and J. A. Shah. Fast scheduling of multi-robot teams with temporospatial constraints. In *RSS*, 2013.
9. P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimal cost paths. *IEEE Trans Sys Sci & Cybernetics*, 4(2), 1968.
10. S. Koenig, P. Keskinocak, and C. Tovey. Progress on agent coordination with cooperative auctions. In *AAAI*, 2010.
11. S. Koenig, C. Tovey, M. Lagoudakis, D. Kempe, P. Keskinocak, A. Kleywegt, A. Meyerson, and S. Jain. The Power of Sequential Single-Item Auctions for Agent Coordination. In *AAAI*, 2006.
12. G. A. Korsah, A. Stentz, and M. B. Dias. A comprehensive taxonomy for multi-robot task allocation. *Int J. Robotics Research*, 32(12), 2013.
13. S. Kraus. Automated negotiation and decision making in multiagent environments. In *MultiAgent Systems and Applications*. 2001.
14. M. Lagoudakis, M. Berhault, S. Koenig, P. Keskinocak, and A. Kelywegt. Simple auctions with performance guarantees for MRTA. In *IROS*, 2004.
15. M. Lagoudakis, V. Markakis, D. Kempe, P. Keskinocak, S. Koenig, A. Kleywegt, C. Tovey, A. Meyerson, and S. Jain. Auction-based multi-robot routing. In *RSS*, 2005.
16. D. Landén, F. Heintz, and P. Doherty. Complex task allocation in mixed-initiative delegation: a UAV case study. In *Principles and Practice of MAS*. 2012.
17. L. Luo, N. Chakraborty, and K. Sycara. Multi-robot assignment algorithm for tasks with set precedence constraints. In *ICRA*, 2011.
18. M. Mataric, G. Sukhatme, and E. Ostergaard. Multi-robot task allocation in uncertain environments. *Autonomous Robots*, 14(2-3), 2003.
19. M. Nanjanath and M. Gini. Repeated auctions for robust task execution by a robot team. *Robotics and Autonomous Systems*, 58(7), 2010.
20. E. Nunes and M. Gini. Multi-robot auctions for allocation of tasks with temporal constraints. In *AAAI*, 2015.
21. A. T. Özgelen, E. Schneider, E. I. Sklar, M. Costantino, S. L. Epstein, and S. Parsons. A first step toward testing multiagent coordination mechanisms on multi-robot teams. In *AAMAS Workshop: ARMS*, 2013.
22. M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop: Open Source Software*, 2009.
23. S. Sariel and T. Balch. Efficient bids on task allocation for multi-robot exploration. In *FLAIRS*, 2006.
24. E. Schneider, O. Balas, A. T. Özgelen, E. I. Sklar, and S. Parsons. Evaluating auction-based task allocation in multi-robot teams. In *AAMAS-ARMS*, 2014.
25. E. Schneider, E. I. Sklar, S. Parsons, and A. T. Özgelen. Auction-based task allocation for multi-robot teams in dynamic environments. In *TAROS*. 2015.
26. A. Schoenig and M. Pagnucco. Evaluating sequential single-item auctions for dynamic task allocation. In *Advances in AI*. 2011.
27. R. G. Smith. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. In *Distributed Artificial Intelligence*. 1988.
28. C. Tovey, M. G. Lagoudakis, S. Jain, and S. Koenig. Generation of Bidding Rules for Auction-Based Robot Coordination. In *Int Multi-Robot Systems Wkshp*, 2005.
29. M. P. Wellman and P. R. Wurman. Market-aware agents for a multiagent world. *Robotics and Autonomous Systems*, 24, 1998.
30. R. Zlot, A. Stentz, M. B. Dias, and S. Thayer. Multi-robot exploration controlled by a market economy. In *ICRA*, 2002.